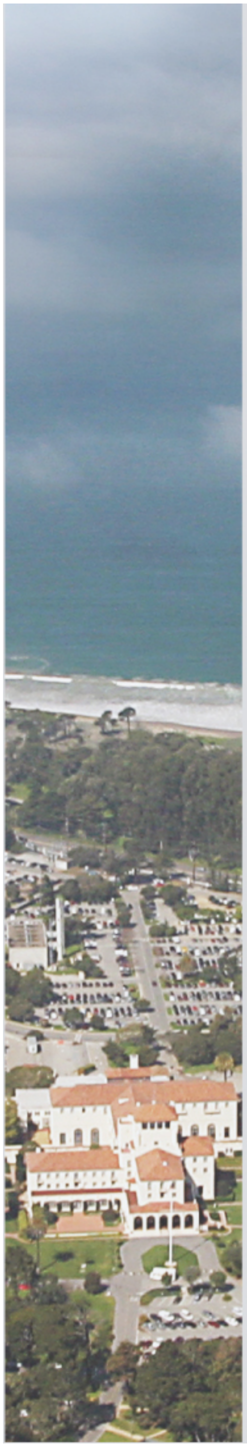


Naval Postgraduate School
Center for Information Systems Security Studies and Research

Assurance Considerations for a Highly Robust TOE

Thuy D. Nguyen, Cynthia E. Irvine, Timothy E. Levin
Department of Computer Science
Naval Postgraduate School

Michael McEvilley
The MITRE Corporation

8th International Common Criteria Conference
Rome, Italy
September 25-27, 2007

Center of Academic Excellence

Monterey, California

HTTP://CISR.NPS.EDU

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **SEP 2007**	2. REPORT TYPE	3. DATES COVERED **00-00-2007 to 00-00-2007**
4. TITLE AND SUBTITLE **Assurance Considerations for a Highly Robust TOE**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School ,Center for Information Systems Security Studies and Research (NPS CISR),Department of Computer Science,Monterey,CA,93943**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
8th International Common Criteria Conference (ICCC), Rome, Italy, 25-27 Sep 2007

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**32**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Discussion Topics

- TOE overview
  - Separation Kernel (SK)
  - Separation Kernel Protection Profile (SKPP)
- Assurance issues for High Robustness
  - Platform Assurance
  - Trusted Initialization
  - Trusted Recovery
- SKPP extended requirements
- Conclusion and plans

Separation Kernel

- Introduced by Rushby (1981)
- Simpler than traditional security kernels
- Primary functional properties
  - Separate system resources into ***security policy equivalence classes***, i.e., ***partitions***
  - Control information flows between and within partitions
- ***Configuration data*** establishes
  - Binding of resources to partitions
  - Policy rules for information flow control
- No support for MAC labels but can be configured to control information flows in a manner consistent with a MLS policy

Least Privilege Separation Kernel

- Refinement of separation kernel
- Apply Principle of Least Privilege to further restrict access to resources
  - Basic SK: homogeneous resource-access requirements
    - Same access authorizations for all subjects in a partition
  - Least Privilege SK: heterogeneous resource-access requirements
    - Separate access authorizations for different subjects in a partition

High Robustness

- **Robustness – US scheme only**
  - Metric for TOE's protection ability
  - Degrees of robustness: Basic, Medium, High
    - Assurance level
    - Strength of security functions
- **Robustness requirement for a TOE**
  - Based on value of data and threats in operational environment
- **High robustness**
  - Provides most stringent protection
  - Can counter sophisticated, well-funded attacks
  - Suitable to protect high value data

Separation Kernel Protection Profile

- U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness
  - Validated in July 2007 (Version 1.03, 29 June 2007)
- Based on Common Criteria Version 2.3
- Assurance requirements
  - Combination of CC-defined components for EAL6 and EAL7
  - Two types of explicitly stated components
    - Modifications of existing CC requirements
    - New requirements

→ No EAL claim due to these extensions

Security Concepts in SKPP

- Enforcement of Partition Information Flow Policy
  – Partition Abstraction, Least Privilege Abstraction
- TOE configuration change
  – Four models: offline, static, constrained, unconstrained
- Establishment of initial secure state
  – Achieved through different degrees of assurance levied on non-TSF components
    - Delivery mechanisms
    - Configuration data generation capability
    - TOE loader
    - Initialization mechanisms
- Trusted recovery
- Platform assurance

Assurance Issues for High Robustness

Platform Assurance

Trusted Initialization

Trusted Recovery

Platform Assurance Issues

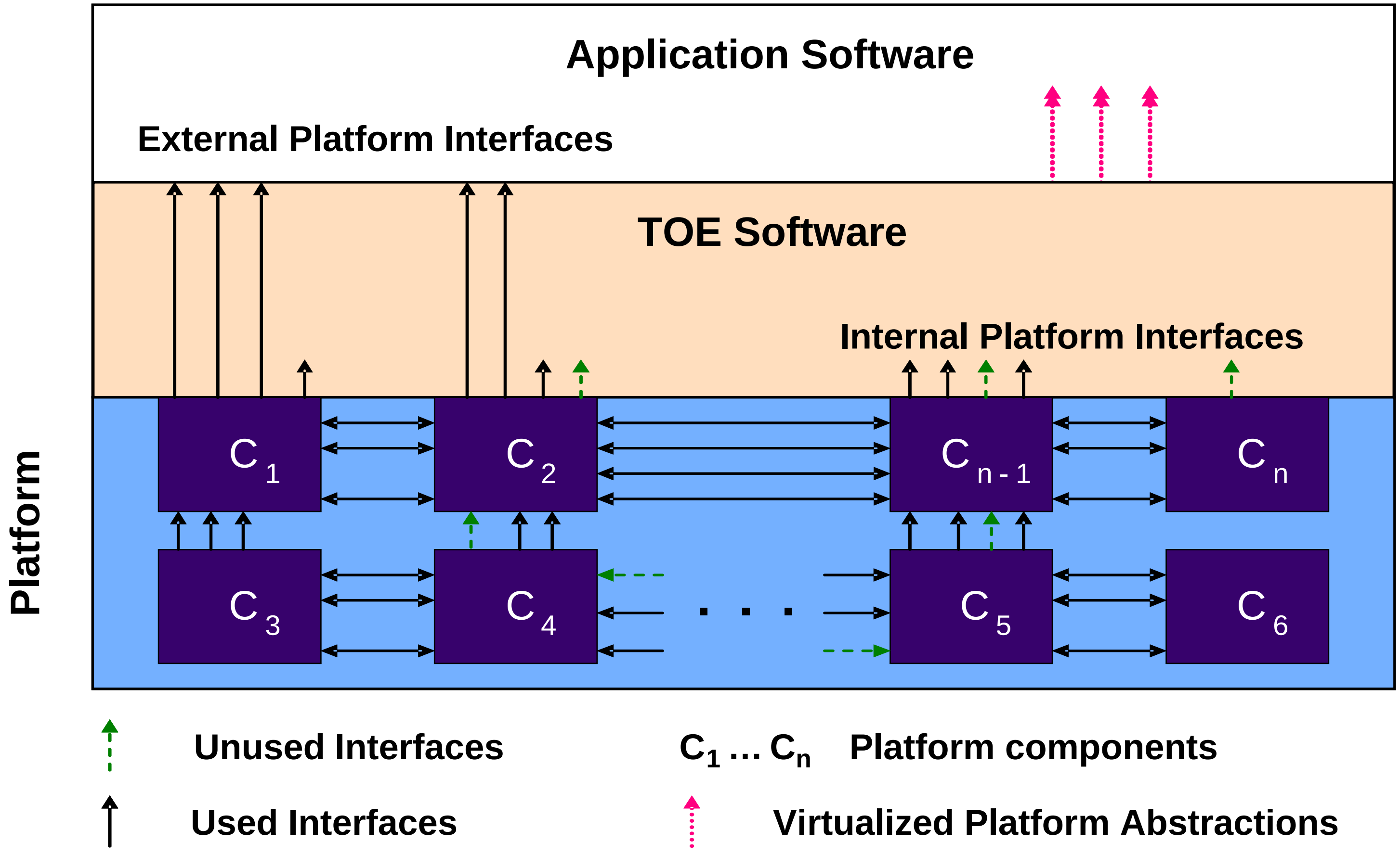
- High robustness requires hardware-supported domain separation and self-protection mechanisms
- No CC-defined requirements for hardware assurance
- Difficult to produce assurance evidence for hardware at same level of detail as software
- Need an assurance framework
  - To assess security properties of hardware mechanisms based on their interfaces to software
  - To establish trust in security-relevant hardware mechanisms
  - To address hardware obsolescence during and after TOE evaluation

→ New Class APT -- Platform Assurance

Platform Concepts

- **Platform = hardware + associated firmware**
- **Platform component**
  - Independently procurable, mass-produced, non-specialized
- **TOE platform = one or more platform components**
  - Defined by ST author
- **Platform definition can vary based on intended usage of the TOE**
  - Very restrictive: require a specific component type with exact properties
  - Less restrictive: allow variations in properties of a specific component type
  - More open: allow use of different component types with defined assembly rules
- **Platform interface**
  - Internal: accessible only to TOE components
  - External: accessible to both TOE components and entities outside the TOE

Hardware/Software Relationships



Trusted Initialization Issues

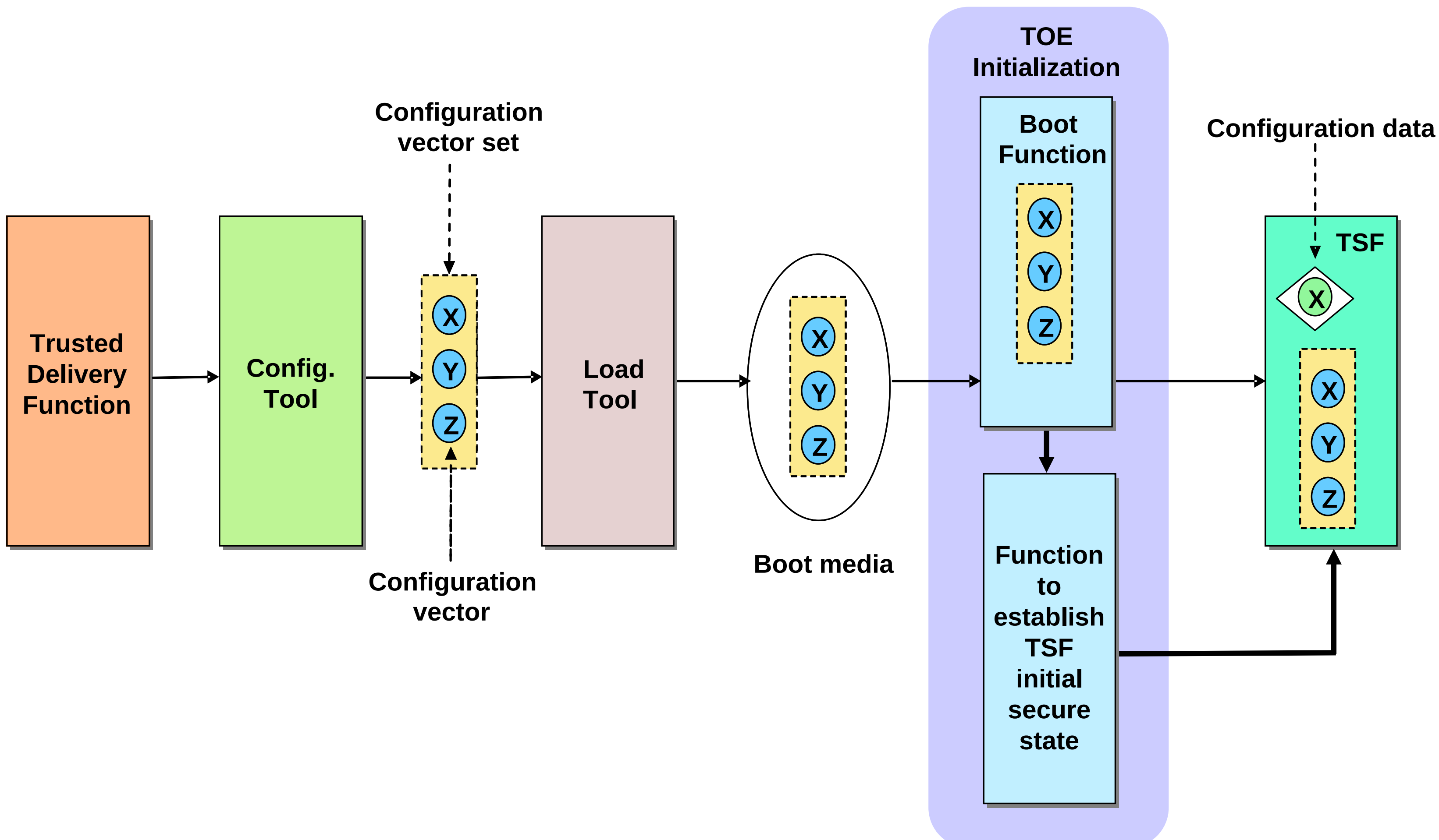
- CC Version 2.x defines no requirements for TOE initialization
  - Rely on administrative actions to ensure proper TOE initialization
- Intended usage of SK requires autonomous TOE initialization
- TSF cannot initialize itself
  - Formal model assumes TSF starts in an initial secure state
- Need a robust mechanism to
  - Establish execution environment for the TSF
  - Bring the TSF to an initial secure state defined by configuration data
- Generation and loading of configuration data need commensurable assurance

SKPP Approach to TOE Initialization

- Correct TOE initialization is achieved through a trust chain of non-TSF functions
  - Delivery
  - Configuration data generation
  - TOE loading
  - Initialization
- Require use of standardized cryptographic algorithms for trusted delivery
  - American National Standards Institute (ANSI)
  - National Institute Standards and Technology (NIST)
- Apply different developmental assurance measures to other initialization-related functions

$\longrightarrow$ New assurance ADV families

TOE Components



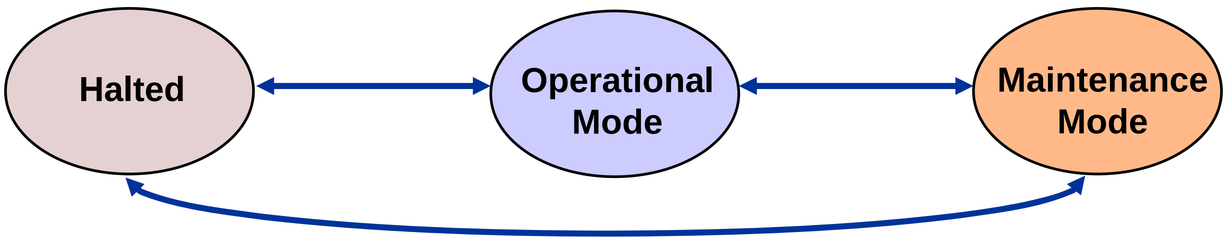
Trusted Recovery Issues

- CC requirements emphasize ways to handle failures and discontinuities
  - Manual versus automated
- CC is vague about presence of recovery functions while in maintenance mode
  - "In the maintenance mode, normal operation might be impossible or severely restricted, as otherwise insecure situations might occur."
- Verification of robustness of recovery mechanisms is difficult
  - Failures/discontinuities have no formal properties

SKPP Approach to Trusted Recovery

- **Focus on protecting the TSF against further compromise during a recovery**
- **Extend FPT_RCV to require the TSF to attempt recovery to a secure state upon detection of an insecure state**
- **Expand definition of maintenance mode**
  - "A contiguous period during an execution session when operational mode functions are restricted, <u>or recovery functions are available that are not available during operational mode, or both</u>."
- **Clarify intended use of maintenance mode**
  - Enable the TOE to return to a secure state
  - Prevent the TOE from entering an insecure state

Maintenance Mode & Secure State



	STATE / MODE	Secure (S)	Insecure (I)
Execution Session	Operational (O)	O\S	O\I
	Maintenance (M)	M\S	M\I
Halted (H)		H\S	n/a

SKPP Extended Requirements

Platform Assurance (APT)

- New assurance class with five families
  - Platform Definition (APT_PDF)
  - Platform Specification (APT_PSP)
  - Platform Conformance Testing (APT_PCT)
  - Platform Security Testing (APT_PST)
  - Platform Vulnerability Assessment (APT_PVA)
- Focus on specifications instead of identifications of components
- Replace a subset of ADV, ATE and AVA requirements for COTS components
  - Specialized components by TOE developer must meet all ADV, ATE and AVA requirements defined for software
- ACM, ADO_DEL and ALC requirements only apply to specialized components
  - Information about CM, delivery, development security are not generally available for COTS components
- Does not address physical protection and anti-tampering issues

Platform Definition (APT_PDF)

- Require Platform Definition Document (PDD) to support component-specific security analysis against SFRs
- PDD can include vendor documentation if they meet content requirements
- PDD include
  - Component types and assembly rules
  - Identification of component interface specifications for all interfaces
  - Security analysis on how each component type interacts with the TOE
  - Precise references to component interfaces so that specifications can be obtained by third-party

Platform Specification (APT_PSP)

- Require complete specifications of platform component interfaces
  - External interface
  - Internal interface
  - Unused interface
- Specifications include
  - Invocation methods, parameters, expected results, error conditions
  - Arguments that all interfaces are included in specifications
- Support functional analysis and vulnerability assessment of the TOE

Platform Conformance Testing (APT_PCT)

- Require functional testing to ensure platform components identified in PDD operate as expected
  - Vendor-provided tests may be used to satisfy this requirement
- Require exercising all security features that are relied upon by the TSF
  - Testing is performed through TSF interfaces
  - Tests are to be developed by TOE developer

Platform Security Testing (APT_PST)

- Require comprehensive security testing
  - Verify correct operations of all external and internal platform interfaces
- Tests to be performed at the component interface level
  - Different than tests in APT_PCT which are at TSF interface level
- Test documentation include
  - Procedures and expected results
  - Argument that test coverage is complete

Platform Vulnerability Assessment (APT_PVA)

- Performed as part of TOE vulnerability analysis
- Assessment is at platform interface level
  - All external platform interfaces
  - All internal platform interfaces used by the TOE
- Complement AVA_VLA requirements
  - Systematic search for vulnerabilities
  - Disposition of identified vulnerabilities
  - Justification that analysis is complete
  - Independent vulnerability analysis by NSA
  - Independent penetration testing by NSA

Trusted Initialization (ADV_INI)

- New family in Class ADV
- Levy both functional and assurance requirements on initialization function
  - Initialization has both testable behaviors and development process
  - SFR paradigm is not applicable to non-TSF components
- Functional responsibilities of initialization function
  - Establish the TSF in an initial secure state
  - Verify integrity of TSF code and data during initialization
  - Handle failures during initialization
  - Provide self-protection during initialization
  - No arbitrary interaction with the TSF after initialization
- Require cooperation from TSF to prevent rogue initialization function
  - Extended SFR requires secure state confirmation by TSF prior to TSP enforcement (FPT_ESS_EXP)

Development Assurance for Initialization

- **Architecture assurance**
  - Self-protection against tampering from other TOE components
  - No interaction with TSF operations after initialization
- **Functional specification**
  - Similar to ADV_FSP requirements for TSF
  - Describe each initialization interface
    - Purpose, method of use, parameters, operations, exceptions, error messages and effects
- **Design documentation**
  - One level of specification, i.e., not as rigorous as ADV_HLD and ADV_LLD for TSF
  - Require modular composition of components
  - Module characterization is based on relevancy to secure state establishment (SSE)
    - SSE-related, SSE-unrelated
- **Test documentation**
  - Test plan, test procedures, expected results, actual results

Configuration Tool Design (ADV_CTD)

- Configuration vector(s) define the initial secure state
  - Corrupted vector could result in unintended TSF operations
- Need robust Configuration Tool to generate and validate configuration vector(s)
- ADV_CTD levies both functional and assurance requirements on Configuration Tool
- Configuration Tool capabilities
  - Generate human-readable form of configuration vectors with clear semantics to allow validation of intended TOE configuration
  - Preserve semantics of data during conversion between human-readable and machine-readable forms of configuration vectors
  - Apply cryptographic seal(s) on generated configuration vector(s)
- Design documentation
  - Explain how to verify correctness and accuracy of generated configuration vector(s)
  - Same level of abstraction and detail required by ADV_HLD

Load Tool Design (ADV_LTD)

- Similar to ADV_CTD
  - Include both functional and assurance requirements
- TOE loading function needs to be robust
  - Part of the chain of trust to establish initial secure state
  - Must maintain integrity of TOE software and configuration vector(s)
- Load Tool capabilities
  - Convert TOE software and configuration vector(s) into a TOE-usable form
  - Preserve integrity of code and data during conversion
- Design documentation
  - Explain the conversion process
  - Same level of abstraction and detail required by ADV_HLD

Trusted Recovery (FPT_RCV)

- Extend base FPT_RCV.2 component
- TSF must attempt recovery to a secure state upon detection of being in an insecure state
  - After completion of TOE initialization
  - During execution session
- TSF must attempt to halt if unable to complete recovery action
  - Transition to maintenance mode may be an acceptable action for certain TOEs
- ST enumerates pair-wise recovery conditions and associated actions
  - Recovery is implementation-specific
- Require assurance evidence that secure state results from the identified action
  - TSF design specifications
  - Administrative guidance documentation
  - Test analysis documentation

Conclusion and plans

- **Assurance considerations for high robustness not sufficient as addressed in CC Version 2.3**
  - Platform assurance, trusted initialization, trusted recovery
- **SKPP explicitly defined SFRs and SARs to address these issues for a separation kernel TOE type**
- **Most of these extended requirements are applicable to other high assurance TOE types**
- **Next step for this PP development team**
  - Development of another high robustness PP for a more complex TOE
    - Leverage SKPP experience to shorten PP engineering time
  - Challenge is to articulate high robustness requirements in CC Version 3.1 context

Acknowledgements

The authors would like to express their appreciation to the NSA SKPP management team and Olin Sibert, without whom this work could not have been completed.

Questions and Contacts

Thuy D. Nguyen

Department of Computer Science
Naval Postgraduate School
Monterey, California, USA

tdnguyen@nps.edu